The department Trust & Security of SURFnet is responsible for both operations and innovation for Trust & Identity and Security domains. We are running the identity federation SURFconext for Dutch Research & Education, incident response team SURFcert, the DNS and mailfilter infrastructure for SURFnet and its members, as well as a number of other services.

Within the team there is quite a lot of expertise on basic internet technology (like DNS(SEC), mail), identity management and identity federations (single sign-on, SAML, OpenID Connect, etc), cryptography, information security, compliance and privacy.

If you're interested in an assignment, please get in touch with us. Preferably, send us an email stating which assignment you would like to do, and what makes you fit for the job! You'll find the main contact person for each assignment in the table below. If you think you have a great idea of your own and it seems to fit with subjects on this page, please get into contact with Michiel Schok and we will assess if we can accommodate your assignment.

### Student assignments

<table>
<thead>
<tr>
<th>Title</th>
<th>Duration</th>
<th>Contact</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid</td>
<td>3-6 months</td>
<td>Michiel Schok</td>
<td>Solid is a new protocol for the decentralized web, developed by Tim Berners Lee and MIT. Solid will enable the situation where you store data independent of the service that wants to access the data, making it so users are the owner of their own data. For education and research this concept is very interesting. You will study the technology behind Solid, install and run a Solid server yourself, and analyse whether and how the concepts behind Solid can be used in identity federations like SURFconext, within the central EduID service, or for research collaboration.</td>
</tr>
<tr>
<td>SAML sessionless</td>
<td>1-4 months</td>
<td>Thijs Kinkhorst, Piete van der Meulen</td>
<td>SURFconext is our federated authentication service (100M logins annually), enabling students and employees to access hundreds of cloud service in a privacy friendly and user friendly way. For this, mainly the SAML 2.0 protocol is used. When a user starts an authentication flow, SURFconext manages a session using cookies. As soon as the user is authenticated successfully, the IdP redirects the user back to SURFconext, after which SURFconext redirects the user back to the cloud service. More and more, technology like cookie blockers and anti tracking measures, can cripple the cookie based approach, blocking (access) to cookies. In this assignment you will dive into how SAML 2.0 within SURFconext currently works, what the role is of sessions and cookies, and analyse possibilities within the protocol to accomplish the same result, but without cookies or possibly without managing sessions (stateless). And by doing that, make authentication for over 1 million users even better and more reliable.</td>
</tr>
<tr>
<td>U2F implementation eSIM</td>
<td>1-3 months (depend on experience with this technology)</td>
<td>Joost van Dijk</td>
<td>U2F tokens are being used as authentication token in 2-factor authentication. Usually these are implemented as hardware token with a USB interface. There exist bindings for NFC and Bluetooth interfaces however. The eSIM (embedded SIM) is meant for machine-to-machine communication (Internet of Things). Advantage of the eSIM is those can be migrated to a new provider over-the-air (as, for instance, in IoT devices it's impossible to physically replace SIM cards). These can also be used as &quot;normal&quot; SIM card in a mobile phone, enabling changing mobile provider without a physical &quot;SIM-swap&quot; (just think of institutions with tens of thousands of student changing from mobile provider). Extra advantage is that the eSIM is not necessarily a property of the mobile provider and it's easier to access content on the card. Just like a normal SIM card, an eSIM is a smartcard, most of the time a JavaCard that can be programmed. This means every mobile device with an eSIM contains a &quot;secure element&quot; which could also be used, for example, as second factor in authentication. In this project a U2F authentication factor is developed based on a JavaCard applet on thee eSIM, in combination with an Android application that on one hand communicates with the U2F applet, and on the other hand for example uses Bluetooth to communicate with a web browser in a Webauthn scenario.</td>
</tr>
<tr>
<td>Project</td>
<td>Duration</td>
<td>Supervisor</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------</td>
<td>--------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SURFconext OIDC/Oauth2 mobile app tutorial</td>
<td>1 month</td>
<td>Joost van Dijk</td>
<td>SURFconext is an identity federation based on SAML 2.0. As many Service Providers don't know SAML but do know OIDC, we also support OIDC. For mobile native applications which often use a server API, OAuth2 is typically used to enable the mobile app to act on behalf of the user. For that, an OAuth2 server is connected as service provider to SURFconext. The combination of involved technologies and systems can easily make the whole situation very complex. In this project a tutorial is developed, focused on developers, to 'SURFconext enable' a simple sample mobile application. Tools to be used are mod-auth-oidc and AppAuth. NB: check myEduData OO-API-app for possible overlap.</td>
</tr>
<tr>
<td>Remote Document Encryption</td>
<td>3 months</td>
<td>Joost van Dijk</td>
<td>ICAO documents (i.e. passports, but also some driving licences) contain a chip carrying cryptographic keys used for authentication. These can be used across the Internet, a technique called Remote Document Authentication (RDA). This is currently used by the Dutch government for DigiD. By using a clever trick these keys can also be used for encryption: Remote Document Encryption (RDE). The idea is that a message can be encrypted based on a public key in such a way that it can only be decrypted with a corresponding private key by someone with physical control over the passport. Filesender is open source software for exchanging files. Filesender distinguishes itself from other file sharing services by supporting (very) large files and end-to-end encryption. For sending encrypted files, the user currently needs an out-of-band key exchange mechanism with the receiver. The aim of this project is to apply Remote Document Encryption in Filesender, such that files can be exchanged securely based on the sender's public key corresponding to the passport of the receiver. This is a joint project with RDW and RU.</td>
</tr>
<tr>
<td>Strong authentication: RADIUS, SAML 2.0 and SURFsecureID</td>
<td>3 months</td>
<td>Pieter van der Meulen</td>
<td>SURFsecureID is the SURFnet service that adds second factor authentication to the authentication provided by SURFconext. SURFsecureID and SURFconext authentication uses the SAML 2.0 SSO protocol, which is a web based protocol. This means that authentication requires the presence of a web browser. We want to enable as much as applications as possible to use the (second factor) authentication of SURFsecureID. Previous efforts resulted in adding “second factor only” (SFO) authentication to SURFsecureID and developing a multi-factor-authentication (MFA) extension for Microsoft AD FS. As a next step we want you to research and build a proof-of-concept for adding a RADIUS server to SURFsecureID. RADIUS is typically used by VPN services and other remote access services. Adding a RADIUS server to SURFsecureID means creating some sort of bridge between the web based SAML 2.0 world and the decidedly non-web server-to-server authentication world of RADIUS. A solution could be building a website where a user can login using their web-based two-factor authentication provided by SURFsecureID where you could get a one-time-password (OTP) to authenticate using the RADIUS client. The idea is simple, but leads to many questions for you to research. E.g.: - How do the applications that SURFsecureID customers want to use, use RADIUS? - What is the security of the solution? How resistant is it against e.g. phishing? - How to make the solutions as user friendly as possible? - Can we optimise the user experience for common second factor types that are used in SURFsecureID, like Tiqr, to completely/partly skip the web based part of the solution? SURFsecureID is uses the opensource software OpenConext-Stepup.</td>
</tr>
<tr>
<td>Zichtbaarheid</td>
<td>3 months</td>
<td>Wim Biemol</td>
<td>SURFcert is voor de zichtbaarheid van incidenten grootdeels afhankelijk van &quot;betrouwbare bronnen&quot;. Bijvoorbeeld een sinkhole waar naartoe een besmet systeem verbinding maakt. Waarschijnlijk missen we door deze wijze wel de nodige incidenten. Het zou mooi zijn als we daar enig inzicht in kunnen krijgen. Door het maken van een vergelijking van een besmet systeem versus de meldingen die daarvoor binnen komen. Is er via die weg een beeld te krijgen wat we zoal kunnen missen en zijn er alternatieven om ook de gemiste incidenten bij SURFcert op de radar te krijgen.</td>
</tr>
</tbody>
</table>
Consent in eduID

SURF is working on eduID, to support lifelong learning, research and collaboration. Students are increasingly taking courses at other institutions. Researchers are collaborating with a number of other research institutions or with companies. SURF is investigating how a single identity, the eduID, can help facilitate this.

An eduID belongs to its user, and the user has a high amount of control over it. This means the user can decide for him/herself which (personal) information to release to which services. One of the biggest challenges for eduID is to design a meaningful system which allows users to control this.

Some users would like more control than others; other users would want to trust the defaults recommended by their institution or another third party. What should such a system look like for these different types of users? Which settings are meaningful to users, which absolutely not?

This assignment is suitable for students from different backgrounds, and even for students who prefer multidisciplinary research:

- Technical: apply and implement the CAR system, developed by Internet2 (a similar organisation as SURF but in the United States) into eduID
- UX design/psychology/information sciences: specify the requirements for eduID and meaningful consent + propose an architecture
- Legal: what should consent look like within eduID with regards to the GDPR + how could CAR (or a similar system) fulfill that role

Federated Appstore

SURFconext is the Dutch national research and education federation, which allows students, teachers, employees and researchers to use their institutional account for logging in to hundreds of online services. On a yearly basis, SURFconext processes more than 100 million logins. Worldwide, eduGAIN interconnects 66 federations with more than 3000 Identity Providers and 2600 services.

A (big) problem for eduGAIN and its federations is the discoverability of services. Both end-users and Identity Provider administrators are often unable to find services that might be of interest for them. For one, this is because there is no single point where these services can be found in a user-friendly manner. There is a technical interface, however, the services’ metadata is often out of date: contact information, service descriptions, logos, etc.

Due to the sheer amount of services, it is impossible for SURF to keep this information up to date. It would be better if the Service Providers themselves supply this information and regularly update this. However, the incentive to do so is currently missing.

To (artificially) create such an incentive, we can borrow ideas from Apple and Google, who’ve demonstrated how to run a succesful Appstore. Wouldn’t it be nice to have something like that for federations as well?

But how should we build it? What should it look like? Would it actually work? For this assignment, there is plenty of room for students with different backgrounds. If you have any ideas, do not hesitate to contact us and inquire about the possibilities!

Honest design for privacy

Privacy is a very hot and relevant topic. Increasingly our lives are supported by digital technologies, and the number of sensors in those technologies keeps on growing. This generates a lot of data, in which companies and governments are very interested, as it can be used to predict us and our behaviour.

Looking at how eager people are to use new technologies and make their lives a little bit more convenient, one could conclude people do not care about privacy at all. This is absolutely not the case. People do care, but they often feel powerless to do something about it or they are simply not aware of the consequences of certain technologies.

Did you know Amazon’s “smart” speaker Alexa is constantly recording everything that is being said and sends it to servers in the US? Most people have no clue. We can solve this by designing products and interfaces which cause “friction”: deliberately do something unexpected. What if Alexa would start talking to you by itself, for instance asking you how your workday was (since it probably knows when you work, where you work, etc.)?

Are you a creative designer with an above average interest in privacy? This assignment is perfect for you.

Real time SURFconext login analysis

SURFconext is our federated authentication service. It is implemented as a central hub, performing more than 100M authentications a year. At busy times, 20 logins a second are handled by the proxy. More than 1000 services are connected to around 200 Identity Providers. Real time logs are available of all authentications passing through.

For this assignment, we’d like you to develop a tool to perform real time analysis of the logins, in order to detect anomalies. The login patterns are very predictable, which makes this project suitable to use machine learning techniques to predict the login statistics. Some of the public statistics can be found here.
pingback:
- TODO: update https://homepages.staff.os3.nl/~delaat/rp/